ELSEVIER

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 29 (2012) 2638 – 2643

Procedia Engineering

www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

Reengineering from Tradition to Cloud: A Case Study

Huang Hexiao[a]*, Zhang Shiming[b], Chen Haijian[a,c],

[a]Department of Information and Engineering, The Open University of Shanghai, Shanghai 200433, China
[b]Shanghai Center for Educational Technology, The Open University of Shanghai, Shanghai 200433, China
[c]School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai 200433, China

Abstract

In China more and more universities are updating their online course system to Cloud-based Education Service, which actually is a re-engineering of online course system. While "Cloud-based Education Service" is not full-fledged, universities have lots of difficulties to deal with in the process of this transition. But what is for sure is that "Cloud-based Education Service" is based on "Could Computing Service." To port the frame and refactor the data access layer of the existing system, we use Google App Engine, Google Web Toolkit and Ext-GWT with service-oriented design. As a result, we successfully port the traditional online course system to Google Cloud Computing Platform. And it is proved that service-oriented developing style Google Cloud Computing Service provided is a scientific way to develop "Cloud-based Education Service". It has the key functions that future "Cloud-based Education Service" requires. Our conclusion is that as Cloud Computing platform supports more and more traditional frameworks, less and less effort will be made to reengineer the online course system.

Keywords: Reengingeering, Cloud Computing, Software Engingeering, Onling Course;

1. Introduction

Nowadays many universities are building "Cloud Based Education Platform" of their own. In 2010, Ministry of Education of China had the "Cloud Based Education Platform" conference in Beijing and Guangzhou. The era of "Cloud Based Education Platform" has come. One of the key tasks is a well-planed migration of existing online course system.

Different from the fact that most of the universities adopt face-to-face education, Shanghai Open University employs online education in most cases. It has more online courses (including model courses)

* Tel.: 86-21-25653393; fax: 86-21-25653393.
E-mail address: hhxgjw@126.com

1877-7058 © 2011 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.proeng.2012.01.364

than general universities, which were developed in versatile developing platforms. Some of them were developed by professional workers with software developing tools, others were developed by teachers themselves by some online course developing tool, such as 'Moodle'[1].

We will discuss the online courses porting of the first category here. Porting to "Cloud Based Education Platform" involves further development, in other words "reengineering".

Reengineering is oriented from software reengineering. Software reengineering is to check, analysis and replace the existing software system in order to refactor it in new style. It also includes the implementation of the new system. Reengineering is an engineering process, it combines reverse engineering, refactoring and forward engineering to rebuild the existing system in a new way. It understands the software (specification, design, and implementation), re-implements the software for new functions, improves its performance or reduces its implementation difficulty. As a result, it achieves the goal to help the software system maintain the present features and get ready for new ones.

Cloud based Education service is one direction, although it is still abstract, it will be realized soon.

"Software Engineering" course is one of online courses we developed before. It has B/S architecture. Developing framework uses Spring + Hibernate + Webwork + EXTJS+JSP. Spring is used for beans' IOC injection. Hibernate is used for database connection and data exchange. Webwork is for page navigation, while EXYJS is an AJAX framework for frame and navigation tree display. Pages are implemented in JSP.

We reengineer the Software Engineering course, and successfully port it to Google Cloud Computing Platform. It is the first attempt for us to port more online courses to Cloud based Education Service. Unfortunately, it is very difficult to port Java projects. The feasibility for Java projects of enterprises level is low. The difficulty was proportional to the number of framework that Google Cloud did not support.

However we find, the way Google App Engine is developed is very different from that in standalone application or cluster application development. It is more service-oriented. The service-oriented way regards service as a basic element, and service provider can register and unregister service dynamically. Service-oriented framework provides event notification: service provider and requester are dynamically bound while running. So the framework is dynamic. With the improvement of Google App Engine SDK, more and more framework will be supported, which can ease reengineering.

2. System Design

2.1. Architecture Design

System architecture is based on Google App Engine. As Google App Engine is distributed, it can not support the uploading of local files (only supports to upload resource files). So we use other web services to host the data required when the system is running. Detail architecture design is shown in Fig. 1.

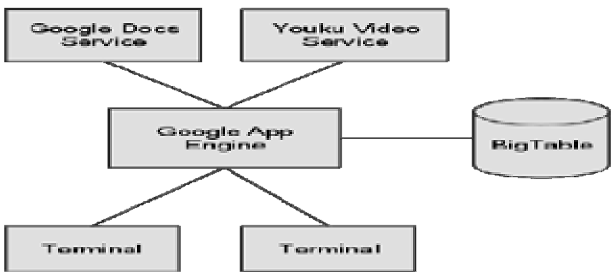

Fig.1. system architecture design diagram

Google Docs Service provides support for docs, presentations and surveys. Youku Video Service provides hosting service for video storage and play (flash based). BigTable is a distributed database system developed by Google. Google App Engine provides Java Persistence to its BigTable storage by a datanuceus-appengine-plugin.

The basic idea to design the architecture is to keep the existing feature if portable. Because Google App Engine does not support local file uploading, “homework uploading” and “grading” features are cancelled. “Students’ Information Management” and “Privilege Management” are merged as “Certificated User Management”, which allows several administrations.

Function module design is shown in Fig. 2.

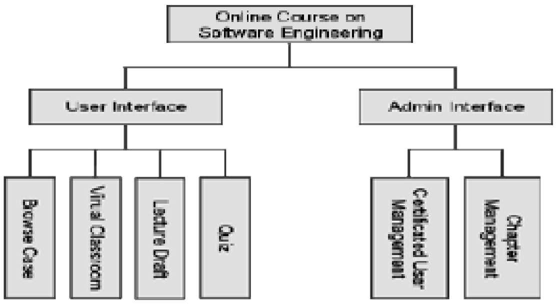

Fig.2. system module diagram

2.2. Sequence Design

User will be asked for his Google account to login the system and be redirected to Google’s authentication page. If he passes Google’s authentication, system will check if that account is in its certificated users list. If that account is in the list, he will be redirected to the home page of the system, where he can browse course information. This is a typical way for students. If that account is an admin, a link to the management console will be added to the homepage. User can manage data and maintain the user list in the backend. System scope diagram is shown below:

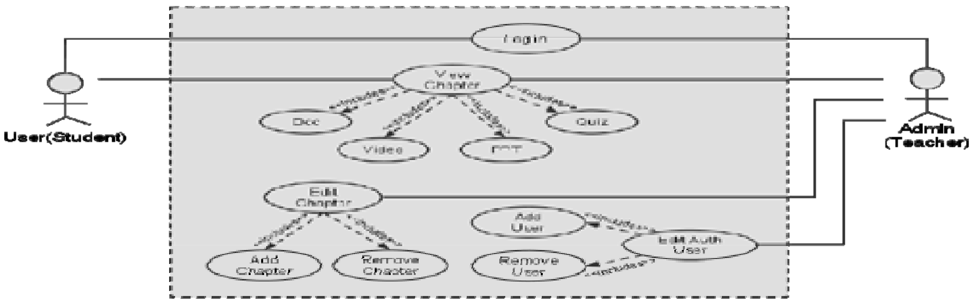

Fig.3. system scope diagram

2.3. Data Persistence Design

Data Persistence is crucial in the system porting. We used MySQL before, but traditional relationship database does not support Google App Engine's storage. We have to rewrite the data access layer and add JDO persistence tag to the old data model in order to support DataNucleus.

Features are changed in the new system, so there are only four types to be modeled not nine. They are User (to store the certificated user list), LoginInfo (to store login information and status), Chapter (to store course information) and ItemInfo (to store tags of each chapter's courseware).

Because LoginInfo is instantiated after login (deleted after logout or timeout), it does not need to be persisted. Class diagram is shown in Fig.4.

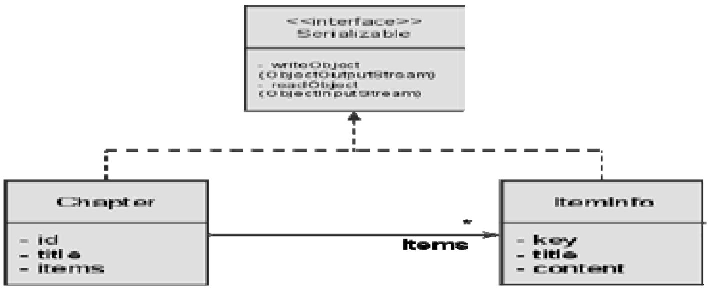

Fig.4. class diagram of chapter

3. Key Feature Design

3.1. User Management

User needs a Google account to login the system, which is authenticated by Google Account Auth service. Then system will check if that account has the privilege to access the system. If this account is valid, system will redirect to the home page of the system. Meanwhile courseware will be loaded on that page. Otherwise user will be rejected to login to the system. If that account is an admin, a link to the management console will be added to the homepage. We fully utilize Google App Engine to valid user's access right after user logins to its Google Account. The reason to employ Google Account Auth is to ease the effort of account authentication and management. Also it improves the security of user accounts.

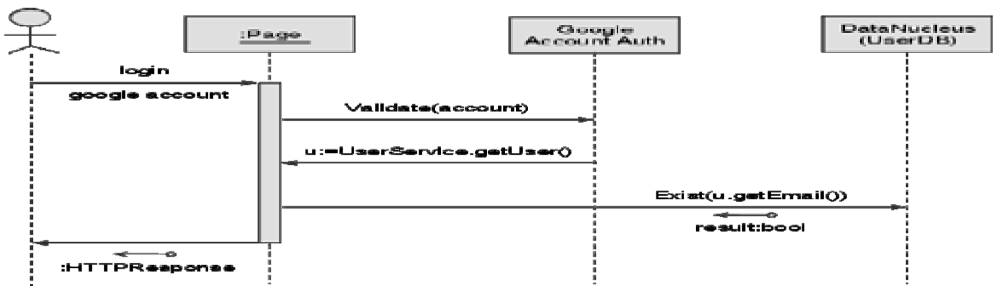

Fig.5. user login sequence diagram

3.2. Chapter Management

After user successfully logins, homepage starts to render every widget on it. The most important task is to fetch the courses' chapters and their sub-element. As data is not hosted on the server, but on corresponding web services, links are actually reference to services. Links' size is small as well as the number of chapters, so we fetch links when loading the homepage instead of getting them after user clicks the chapter title. In this way, user experience is improved.

Besides, chapter content is getting by remote procedure call (RPC) in Google Web Toolkit. It is asynchronous. So chapter content fetching, other query and homepage rendering can be conducted simultaneously. Homepage does not need to refresh to get the data. Sequence diagram is shown in Fig.6.

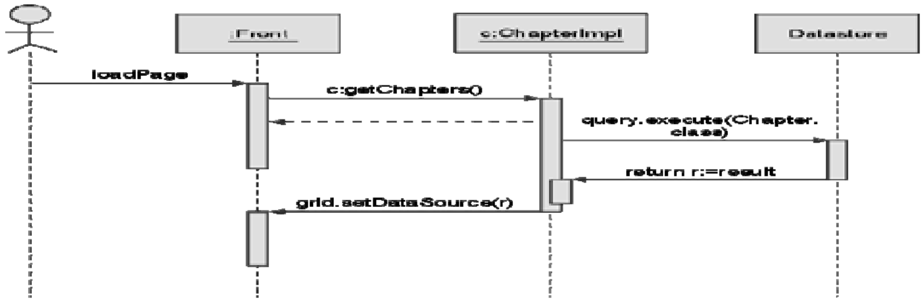

Fig.6. sequence diagram

3.3. User Interface of System

System has front end and back end interface. After user login to the front end homepage, system starts to render every widget, it includes: user information bar, chapter navigation bar and content tab. User name (not Google account name) is shown on the top of the page, as well as the link to backend (if user is a admin) and logout link. On the left, there is chapter navigation bar. User can use "expand" and "collapse" to view chapters. Content of each chapter is shown by tabs on the right. After clicking the chapter title, the first content will be shown on the tab. Fig.7 shows an online test as a piece of courseware.

Fig.7. unit test page

4. Conclusion

In the old system, we used data access layer provided by Hibernate to validate user. Although Hibernate offers easy object-relationship mapping, it still requires extra effort to develop. With Google App Engine, developer can get rid of user authentication process developing, and focus on other tasks.

Besides, we used ExtJS to render before. Several modules were working together to get chapter content. Webwork was used to get data. For Google App Engine does not support Webwork, we adopt Google Web Toolkit and Ext-Gwt to redeveloper this feature.

During the progress of reengineering, we find that Google App Engine provides not only strong computing power but also valuable cloud services on its platform, which standalone/cluster server cannot beat.

Through this project, we find it is very difficult to port Java project to Google Cloud Computing Platform, the feasibility of enterprise level project porting is low. It is related to frameworks in the existing system that Google does not support. Some frameworks in Java project cannot be directly support by Google, or they are not supported at all. So reengineering involves a lot of rewriting. It has a big working load.

Although Google Cloud Computing Platform has its imperfectness (storage, data management and coding framework), it still can effectively fulfill the business requirements and dramatically reduce developing cost and running expense, as it is designed for some specific services [5].

As a conclusion, Google Computing Platform is a scientific choice for service-oriented developing. It has key functions that Cloud based Education Platform asks for. Architecture and data access layer are redesigned, while the course design remains the same. With Cloud Computing Platform supports more frameworks, the porting effort will be less and less.

Acknowledgements

We thank to support from The Open University of China (Grant No.GAQ1307) and Shanghai Association of Higher Education (Grant No.ZCGJ19-10).

References

[1] Wang Jide, Zhang Li. New Progresses in the Domestic Research on Moodle. *Journal of Distance Education*. 2009(5):15-18, 32

[2] Robert S. Arnold. *Software Reengineering.* IEEE Computer Society Press, 1993.

[3] Huang Hexiao et al. Design and Development of Network Course on Software Engineering. *Computer Education.* 2009([3] 22):93-96

[4] Wang Tao, Wei Jun. Component-based Software Reengineering Based on OSGi. *Journal of Frontiers of Computer Science & Technology.* 2009(6):601-611

[5] Sun Jian, Jia Xiaojing. A Study of the Influence of Technical Architecture on the Total Cost of Google Cloud Computing Platform. *Telecommunications Science.* 2010(1):38-44